**JavaScript Data Types**

JavaScript Strings

A string (or a text string) is a series of characters like "John Doe". Strings are written with quotes. You can use single or double quotes

var carName1 = "Volvo XC60";   // Using double quotes  
var carName2 = 'Volvo XC60';   // Using single quotes

## JavaScript Numbers

JavaScript has only one type of numbers. Numbers can be written with, or without decimals

var x1 = 34.00;     // Written with decimals  
var x2 = 34;        // Written without decimals

## JavaScript Booleans

Booleans can only have two values: true or false.

var x = 5;  
var y = 5;  
var z = 6;  
(x == y)       // Returns true  
(x == z)       // Returns false

## JavaScript Arrays

JavaScript arrays are written with square brackets. Array items are separated by commas.

var cars = ["Saab", "Volvo", "BMW"];

## JavaScript Objects

JavaScript objects are written with curly braces {}. Object properties are written as name:value pairs, separated by commas.

var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

## Undefined

In JavaScript, a variable without a value, has the value undefined. The type is also undefined.

var car;    // Value is undefined, type is undefined car=undefined;    // Value is undefined, type is undefined

## Empty Values

An empty value has nothing to do with undefined. An empty string has both a legal value and a type.

var car = "";    // The value is "", the typeof is "string"

## Null

In JavaScript null is "nothing". It is supposed to be something that doesn't exist. Unfortunately, in JavaScript, the data type of null is an object.

var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};  
person = null;    // Now value is null, but type is still an object

var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};  
person = undefined;   // Now both value and type is undefined

## Difference Between Undefined and Null

undefined and null are equal in value but different in type:

typeof undefined           // undefined  
typeof null                // object  
  
null === undefined         // false  
null == undefined          // true

## The typeof Operator

You can use the JavaScript typeof operator to find the type of a JavaScript variable. The typeof operator returns the type of a variable or an expression:

typeof ""             // Returns "string"  
typeof "John"         // Returns "string"  
typeof "John Doe"     // Returns "string"

typeof 0              // Returns "number"  
typeof 314            // Returns "number"  
typeof 3.14           // Returns "number"  
typeof (3)            // Returns "number"  
typeof (3 + 4)        // Returns "number"

typeof {name:'John', age:34} // Returns "object"   
typeof [1,2,3,4]             // Returns "object" (not "array", see note below)  
typeof null                  // Returns "object"  
typeof function myFunc(){}   // Returns "function"